GNU/Linux – Eine Einführung

Computer- und Mediensicherheit
FH Hagenberg
Wintersemester 2004/2005

Agenda

- Befehlsausführung
- Rechte und Benutzer
- Grundlegende Befehle
- Ausgabe- und Kommanodumleitung
- SSH Suite
- Mounting
- Information über das System

Programme ausführen

Programme ausführen

- einfach Programmnamen eintippen
- dazu muss Verzeichnis in PATH sein
  - spezielle Umgebungsvariable
  - analog zu Windows
- sonst Pfadangabe erforderlich
  - /usr/bin/less – absoluter Pfad
  - bin/less – relativer Pfad

Hilfe

less

- zeigt Text Bildschirmweise an
- Aufruf: `less` *`datei`*
- wichtige Tastenkombinationen
  - `q` Beenden (*quit*)
  - `<space>` einen Bildschirm weiter
  - `<enter>` eine Zeile weiter
  - `g` ganz an den Anfang
  - `G` ganz ans Ende
  - `/blah` nach "blah" suchen
  - `n` nächstes Vorkommen von "blah"

Manpages

- Dokumentation zu Programmen
- Aufruf: `man` *`begriff`*
- benutzt less
- unterteilt in Sektionen
  - bei zwei gleichen Begriffen, die unterschiedliche Manpages produzieren
  - in diesem Fall: `man` *`sektion begriff`*
  - `apropos` *`begriff`* verrät Sektionen

info

- GNU-Leute schreiben lieber darin ihre Dokumentation
- Aufruf: `info` *`begriff`*

- unterteilt in Nodes
  - `n` nächtse Node
  - `p` vorherige Node
  - siehe auch `info info`

vi – Ein Texteditor

vim – visual editor (improved)

- Texteditor aus den 70ern
- überraschenderweise immmer noch populär
- gewöhnungsbedürftig
- zwei Modi
  - Kommandomodus
  - Editiermodus
- Bedienung siehe Reference Cards

vi – nützliche Kommandos

- *x* yy – *x* Zeilen kopieren
- *x* dd – *x* Zeilen löschen
- p – unter aktueller Zeile einfügen
- P – über aktueller Zeile einfügen
- */begriff* – nach *begriff* suchen
  - n – zum nächsten Ergebnis springen
- 1,$ s/*suche*/*ersetze*/g

Umgang mit Dateien und Verzeichnissen

Dateinamen

- Länge von Dateinamen: 255 Zeichen
- Erlaubte Zeichen: alle außer '/'
- Unterscheid zwischen Groß/Kleinschreibung
- Punkt im Namen
  - Teil des Namens, bessere Lesbarkeit
  - Extension teilw. ausgewertet, z.B. C-Compiler
  - Punkt am Beginn des Namens → "versteckte Dateien"

ls

- zeigt Verzeichnisinhalt an
- Optionen
  - `-l` langer Output
  - `-a` zeige auch versteckte Dateien
  - `--color` buunt
  - `-h` Größe in kByte/MByte/etc.

Im Verzeichnisbaum navigieren

- `pwd` – gegenwärtiges Verzeichnis
- `cd` – change directory
  - `cd ..` – Überverzeichnis
  - `cd /home/cms0101` – absoluter Pfad
  - `cd cms01018` – relativer Pfad
  - `cd -` – zurück zum letzten
- `mkdir` – Verzeichnis erstellen
- `rmdir` – Verzeichnis löschen

Dateien verwalten

- `cp` – copy
- `mv` – move
- `rm` – remove
- `touch` – berühren oder anlegen
- `cat` – ausgeben
- `less` – anzeigen

Sonderzeichen

?	genau ein beliebiges Zeichen
*	beliebig viele (auch null) beliebige Zeichen
[abc]	genau eines der angegebenen Zeichen
[a-f,x-z]	ein Zeichen aus dem angegebenen Bereich
[!abc]	keines der angegebenen Zeichen
[^abc]	wie [!abc]
~	Home-Verzeichnis

Sonderzeichen

- Aufhebung der Interpretation

\	hebt Interpretation für das nächste Zeichen auf
'abc'	Aufhebung der Interpretation aller eingeschlossenen Zeichen
"abc"	Aufhebung der Interpretation aller eingeschlossenen Zeichen bis auf $, \ und `

Zugriffsrechte

Zugiffsrechte

- Unix verwendet eigentümerbasierte Zugriffskontrolle
- stellt sicher, dass
  - jede Ressource im System einen Eigentümer hat
  - Eigentümer alle Rechte auf die Ressource hat
  - er anderen Benutzer Rechte gewähren oder entziehen darf
- Unix hat darüber hinaus einen speziellen User – Superuser oder root
  - automatisch Eigentümer aller Ressourcen
  - dem Superuser können keine Rechte entzogen werden

Zugriffsrechte

- Schutz privater Dateien vor Zugriff anderer Systembenutzer
- Durchsetzung der Rechte wird vom Betriebssystem garantiert

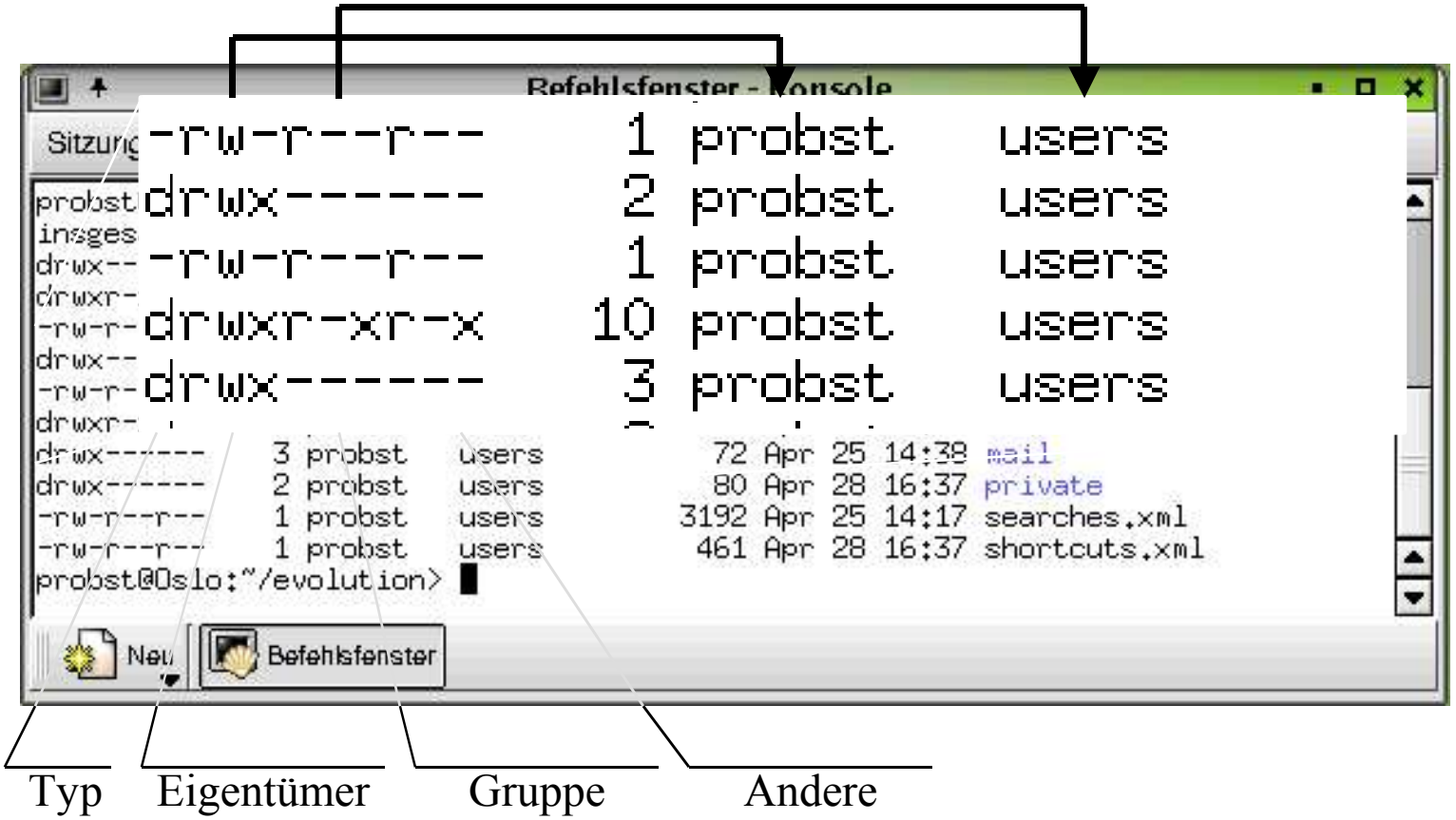
Granularität und Auswertung

- Granularität der Rechte
  - Eigentümer (user)
  - Benutzergruppe (group)
  - Alle anderen Anwender (others)
- Auswertung der Rechte
  1. Prüfen der Eigentümerschaft
  2. Prüfen der Zugehörigkeit zur Gruppe
  3. sonst Zugehörigkeit zu anderen Anwendern

Benutzer – Gruppenzuordnung

- Jeder Benutzer gehört mindestens einer (Haupt-)Gruppe an
- Kann mehreren Nebengruppen zugeordnet werden
- Eigentümer kann Rechte für eine Gruppe oder für andere vergeben

Rechte anzeigen



Rechteübersicht

Recht	***Datei***	***Verzeichnis***
r (read)	Datei kann **gelesen** werden	Verzeichnisinhalt kann **angezeigt** werden
w (write)	Datei kann **geschrieben** werden	Verzeichniseintrag kann **erstellt** und **gelöscht** werden
x (execute)	Datei kann **ausgeführt** werden	In Verzeichnis kann **gewechselt** werden

Rechtevergabe

- Eigentümer oder root können Rechte vergeben/ändern
- können relativ oder absolut vergeben werden
  - relativ – bestehende Rechte erweitert
  - absolut – bestehende Rechte überschrieben
- symbolischer und direkter Modus

Rechtevergabe symbolisch

- `chmod [Bereich] Operand Berechtigung datei`

Bereich	u Eigentümer (user) g Gruppe (group) o Übrige Benutzer (others) Keine Angabe = ugo
Operand	+ Recht hinzufügen - Recht wegnehmen = Recht absolut setzen
Berechtigung	r Read w Write x eXecute

Rechtevergabe symbolisch

`chmod +x` *`datei`*	Execute-Recht für Benutzer, Gruppe, Andere
`chmod go-w` *`datei`*	Gruppe und Andere Schreibrecht entziehen
`chmod g+rwx` *`d`*	Alle Rechte für die Gruppe
`chmod ugo-rwx` *`d`*	Alle Rechte für alle entziehen
`chmod u-r,g+x` *`d`*	Benutzer Schreibrecht entziehen und Gruppe Execute-Recht zuweisen
`chmod u=rw` *`d`*	Schreib- und Leserecht für Benutzer

Rechtevergabe direkt

- `chmod` *`modus dateiname`*
- `modus` sind 3 Oktalzahlen
  - Jede Stelle steht für Bereich
    1. User
    2. Group
    3. Others
- zugeteiltes Recht ergibt sich aus Summe der Einzelrechte

  rwx = read + write + execute ➔ 4+2+1 = 7

Recht	Zahl
read	4
write	2
execute	1

Weitergabe der Berechtigungen

- Eigentümer
  - Benutzer wird neuer Eigentümer, alter Eigentümer verliert Rechte an Datei
  - chown *username dateiliste*
  - Befehl kann nur vom Superuser ausgeführt werden
- Gruppenzugehörigkeit
  - chgrp *gruppenname dateiliste*
  - Befehl kann nur vom Superuser oder vom Eigentümer ausgeführt werden, sofern dieser auch Mitglied der neuen Gruppe ist

Benutzerverwaltung

Daten über Benutzer

- `/etc/passwd` – Basisdaten
- `/etc/shadow` – Passwörter
- `/etc/group` – Gruppen

- in Dateisystem nur numerische UIDs gespeichert
  - Auflösung erfolgt durch Programme
- Passwörter in eigener Datei, da `/etc/passwd` von jedem einsehbar

/etc/passwd

- Daten in einer Zeile, getrennt durch ':'
- Login-Name
- Passwort, bei modernen Systemen ein x
- User ID (0 - 50 000)
- Group ID (0 - 50 000)
- Kommentar (z.B. Name des Benutzers)
- Home-Verzeichnis
- Programm, das gestartet werden soll (Shell)

/etc/shadow

- Name
- Passwort
- Letzte Änderung (ab 1.1.1970 in Tagen)
- Minimale Gültigkeitsdauer (in Tagen)
- Maximale Gültigkeitsdauer (in Tagen)
- Vorwarnzeit (Anzahl in Tagen)
- Inaktiv (Zeitspanne in Tagen)
- Verfall (absolute Datumsangabe)
- Kennzeichen (reserviert)

/etc/group

- speichern Gruppen und Mitglieder
- Hauptgruppe eines Benutzers aber in /etc/passwd

Benutzer anlegen/verwalten

- `useradd` *`username`*
- `userdel` *`username`*
- `usermod` *`username`*

- `groupadd` *`groupname`*
- `groupdel` *`groupname`*

Passwörter

- `passwd` ändert Passwort für eigenen User
  - Superuser kann auch andere Passwörter ändern: `passwd` *`username`*
- Superuser kann Accounts sperren `passwd –l` *`username`*
- `passwd –u` *`username`* hebt Sperre auf

Benutzerinformation

- `whoami` – Wer bin ich?
- `who` – Wer ist gerade eingeloggt?

Befehlsumleitung

Was ist Befehlsumleitung?

- manchmal will man die Ausgabe eines Befehls in eine Datei oder woanders hin umleiten
- manchmal will man die Eingabe eines Befehls aus einer Datei lesen
- manchmal will man die Ausgabe eines Befehls an einen anderen füttern
- das geht unter Unix ganz leicht

Ein bisschen Unix-Internals

- 3 Standard-Filedescriptoren
  - Standardeingabe `stdin`
    - meist Tastatur
  - Standardausgabe `stdout`
    - meist Bildschirm
  - Standardfehlerausgabe `stderr`
    - meit ebenfalls Bildschirm
    - Trennung im Hinblick auf Umleitung sinnvoll

Standardfiledescriptoren

- sind automatisch offen
- viele Funktionen greifen automatisch auf diese zu

- diese können nun (fast) beliebig umgeleitet werden
  - in Dateien
  - an andere Programme

Ein bisschen Unix-Internals

- dadurch Kommandos potentiell einfacher
  - müssen nur von `stdin` lesen können
  - müssen nur an `stdout` ausgeben können

- viele Kommandos können trotzdem noch extra mit Dateien umgehen

Ausgabeumleitung

- Ergebnis wird in Datei geschrieben, nicht auf Terminal
- Umleitung erfolgt durch '>' bzw. '>>'
  - '>' fängt von vorn an, '>>' hängt an
- *kommando > dateiname*
- *kommando >> dateiname*

- `cat /etc/passwd > passwd.kopie`

Eingabeumleitung

- Programm liest aus Datei, nicht von Tastatur
- Umleitung erfolgt durch '<'
- *kommando < dateiname*

- `less < /etc/passwd`

Umleitung der Fehlerausgabe

- Fehlermeldungen werden in Datei geschrieben
- Umleitung erfolgt durch '2>'
- *kommando 2> dateiname*

- Kombinationen
  - *kommando < eingabedat > ausgabedat 2> fehlerdat*

Spezialziele

- *Everything is a File*
- daher Spezialziele/-quellen
  - `/dev/null` – schwarzes Loch für arme Bits; nur Ziel
  - `/dev/zero` – lauter Null-Bytes ('0x00')
  - `/dev/random` – Zufallsbytes
  - praktisch jede beliebige andere Gerätedatei möglich

Pipes

- verbindet Ausgabe eines Programms mit Eingabe eines zweiten Programms über temporären Puffer
- Mehrere Kommandos können hintereinander geschaltet werden
- *kommando* | *kommando2* | *kommando3*

- `ls –la | less`

Remotetools

Remotetools

- bauen eine Verbindung zum PC über das Netzwerk auf
- heute meist verschlüsselte Protokolle
- ermöglichen eine Administration des Rechners ohne lokalen Zugang

- früher hauptsächlich `telnet`
  - dummerweise unverschlüsselt

SSH – Secure Shell

- verbreitetste Remote Shell
- baut Verbindung verschlüsselt auf
  - in Version 2 sicher gegen Man-in-the-Middle Attacken

- `ssh` *`benutzername`*`@`*`zielrechner`* [*`kommando`*]

SCP – Secure Copy

- Teil des SSH-Pakets
- macht sichere Dateitransfers möglich
- funktioniert in beide Richtungen

- `scp [user@quelle:]/pfad [user@ziel:]/pfad`

SSH – Schlüsselbasierte Authentisierung

- basiert auf Public-Key-Verfahren
- es wird keine geheime Information übertragen
- `ssh-keygen` erzeugt Schlüsselpaar
- öffentlicher Schlüssel muss auf entfernten Rechern bekannt sein

SSH – X Forwarding

- X ist netzwerkfähig
  - Programme auf entferntem Rechner leiten Ausgabe auf lokalen Schirm
  - benötigt X-Server auf Empfängerseite
  - Verbindung aber ohne Authentisierung
- SSH ermöglicht sichere Weiterleitung der GUI-Ausgabe
  - Authentisierung
  - Verschlüsselung

SSH – Port Forwarding

- kann beliebige TCP-Ports weiterleiten
- Anwendungsgebiete
  - Verschlüsselung für Klarttextprotokolle
  - Umgehung von Portrestriktionen

- `ssh -L port:host:hostport`
- `ssh -R port:host:hostport`

Übung

1. Konfiguration des Server sichern
   1. nur Protokollversion 2 erlauben
   2. Login für Superuser nicht gestatten
   3. Service neu starten

3. Spaß mit dem Rechner des Nachbarn
   - `ssh cms03@`*`zielrechner`*` eject`

Mounten

Warum mounten?

- beim Booten des Systems wird /etc/fstab abgearbeitet
- dort steht wohin die erstellten Partitionen gemountet werden

```
/dev/hda1    /boot     ext2   noauto,noatime     1 1
/dev/hda3    /         ext3   noatime           0 0
/dev/hdb1    /home     ext3   noatime           0 0
```

Wie wird gemountet

- `mount –t` *`typ quelle ziel`*
- *`umount quelle oder umount ziel`*

- Wichtig bei Disketten: Vor dem entfernen immer unmounten, da sonst die Veränderungen nicht auf die Diskette geschrieben werden.

Was kann man mounten?

- CD-ROM
- Diskette
- Dateien mit Filesystemen (CD-ISOs)
- USB-Sticks
- entfernte Dateisysteme (Netzlaufwerke)

Disktools

nützliche Hilfsprogramme

- `df` – freier Plattenspeicher
- `du` – belegter Plattenspeicher
- `locate` – findet Dateien mit Datenbank
- `updatedb` – aktualisiert DB für locate
- `find` – findet Dateien (langsam)
- `which` – findet den Pfad zu ausführbaren Dateien

Prozesse

Prozesse

- jeder Prozess hat eindeutig ID
  - bleibt ihm sein Leben lang erhalten
  - Prozess 1 ist init-Prozess

- jeder Prozess (außer init) hat einen "Vater" – jener Prozess, der ihn aufgerufen hat

Informationen über Prozesse

- `ps`
  - Snapshot der gerade laufenden Prozesse
  - viele verwirrende Optionen
  - `ps auxf` meist ausreichen
- `top`
  - Prozessinformationen interaktiv und vollbild
  - liefert auch CPU- und Speichernutzung

Prozesse beenden

- `kill -`*`SIGNAL PID`*
  - sendet Signal an Prozess
    - `KILL` – sofort beenden
    - `TERM` – ordentlich beenden
    - `HUP` – spezielle Bedeutung bei Serverp.

- `killall` *`name`*
  - sendet Signal an alle Prozesse mit Namen *`name`*

Statusinformationen

System allgemein

- `uname`
  - zeigt Systemtyp, Kernel inkl. Version
- `uptime`
  - wie lange das System ohne Unterbrechung läuft
  - Auslastung des Systems

Hauptspeicher

- free
  - belegter/freier Hauptspeicher
  - wieviel davon für Festplattencache genutzt wird

```
             total       used       free     shared    buffers     cached
Mem:         62080      61304        776          0       6132      43488
-/+ buffers/cache:      11684      50396
Swap:       155224         16
```

Das /proc Pseudodateisystem

- Information über laufende Prozesse
  - `/proc/`*`PID`*`/cmdline` – Kommandozeile
  - `/proc/`*`PID`*`/exe` – Link auf Datei
- Informationen über den Kernel
  - `/proc/version` – Versionsnummer
- Informationen über das System
  - `/proc/cpuinfo`
  - `/proc/meminfo`
- Einstellungen für den Kernel

Netzwerk

- `netstat`
  - zeigt aktive Verbindungen an
  - `netstat –t` – nur TCP
  - `netstat –u` – nur UDP
  - `netstat –l` – listening Ports (Server)

- `arp`
  - welche ARP-Adresse gehört zu welcher IP

sonstige Fragen und/oder Anmerkungen